On Mining Complex Sequential Data by Means of FCA and Pattern Structures

Aleksey Buzmakov, Elias Egho, Nicolas Jay, Sergei O. Kuznetsov, Amedeo Napoli, Chedy Raïssi

To appear in the *International Journal of General Systems*
Vol. 00, No. 00, Month 20XX, 1–23

On mining complex sequential data by means of FCA and pattern structures

Aleksey Buzmakov[a,c*], Elias Egho[b,1], Nicolas Jay[a], Sergei O. Kuznetsov[c], Amedeo Napoli[a] and Chedy Raïssi[a]

[a] *Orpailleur, LORIA (CNRS – Inria NGE – University of Lorraine), Vandœuvre-lès-Nancy, France*; [b] *Orange Labs, Lannion, FRANCE* [c] *National Research University Higher School of Economics, Moscow, Russia*

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

Nowadays data sets are available in very complex and heterogeneous ways. Mining of such data collections is essential to support many real-world applications ranging from healthcare to marketing. In this work, we focus on the analysis of *"complex"* sequential data by means of interesting sequential patterns. We approach the problem using the elegant mathematical framework of Formal Concept Analysis (FCA) and its extension based on *"pattern structures"*. Pattern structures are used for mining complex data (such as sequences or graphs) and are based on a subsumption operation, which in our case is defined with respect to the partial order on sequences. We show how pattern structures along with projections (i.e., a data reduction of sequential structures), are able to enumerate more meaningful patterns and increase the computing efficiency of the approach. Finally, we show the applicability of the presented method for discovering and analyzing interesting patient patterns from a French healthcare data set on cancer. The quantitative and qualitative results (with annotations and analysis from a physician) are reported in this use case which is the main motivation for this work.

Keywords: data mining; formal concept analysis; pattern structures; projections; sequences; sequential data

1. Introduction

Sequence data is present and used in many applications. Mining sequential patterns from sequence data has become an important data mining task. In the last two decades, the main emphasis has been on developing efficient mining algorithms and effective pattern representations (Han et al. 2000; Pei et al. 2001b; Yan, Han, and Afshar 2003; Ding et al. 2009; Raïssi, Calders, and Poncelet 2008). However, one problem with traditional sequential pattern mining algorithms (and generally with all pattern enumeration algorithms) is that they generate a large number of frequent sequences while a few of them are truly relevant. To tackle this challenge, recent studies try to enumerate patterns using some alternative interestingness measures or by sampling representative patterns. A general idea in finding *statistically significant patterns* is to extract patterns whose characteristics for a given measure, such as frequency, strongly deviates from its expected value under a null model, i.e. the value

[1] Elias Egho was in LORIA (Vandœuvre-lès-Nancy, France) when this work was done.
*Corresponding author. Email: aleksey.buzmakov@inria.fr

expected by the distribution of all data. In this work, we focus on complementing the statistical approaches with a sound algebraic approach trying to answer the following question: *can we develop a framework for enumerating only relevant patterns based on data lattices and its associated measures?*

The above question can be answered by addressing the problem of analyzing sequential data using the framework of Formal Concept Analysis (FCA), a mathematical approach to data analysis (Ganter and Wille 1999), and pattern structures, an extension of FCA that handles complex data (Ganter and Kuznetsov 2001). To analyze a dataset of "complex" sequences while avoiding the classical efficiency bottlenecks, we introduce and explain the usage of projections, which are mathematical mappings for defining approximations. Projections for sequences allow one to reduce the computational costs and the volume of enumerated patterns, avoiding the infamous "pattern flooding". In addition, we provide and discuss several measures, such as stability, to rank patterns with respect to their "interestingness", giving an expert order in which the patterns may be efficiently analyzed.

In this paper, we develop a novel, rigorous and efficient approach for working with sequential pattern structures in formal concept analysis. The main contributions of this work can be summarized as follows:

- *Pattern structure specification and analysis.* We propose a novel way of dealing with sequences based on complex alphabets by mapping them to pattern structures. The genericity power provided by the pattern structures allows our approach to be directly instantiated with state-of-the-art FCA algorithms, making the final implementation flexible, accurate and scalable.
- *"Projections" for sequential pattern structures.* Projections significantly decrease the number of patterns, while preserving the most interesting ones for an expert. Projections are built to answer questions that an expert may have. Moreover, combinations of projections and concept stability index provide an efficient tool for the analysis of complex sequential datasets. The second advantage of projections is its ability to significantly decrease the complexity of a problem, saving thus computational time.
- *Experimental evaluations.* We evaluate our approach on real sequence dataset of a regional healthcare system. The data set contains ordered sets of hospitalizations for cancer patients with information about the hospitals they visited, causes for the hospitalizations and medical procedures. These ordered sets are considered as sequences. The experiments reveal interesting (from a medical point of view) and useful patterns, and show the feasibility and the efficiency of our approach.

This paper is an extension of the work presented at CLA'14 conference (Buzmakov et al. 2013). The main differences w.r.t. the CLA'14 paper are a more complete explanation of the mathematical framework and a new experimental part evaluating different aspects of the introduced framework.

The paper is organized as follows. Section 2 introduces formal concept analysis and pattern structures. The specification of pattern structures for the case of sequences is presented in Section 3. Section 4 describes projections of sequential pattern structures followed in Section 5 by the evaluation and experimentations. Finally, related works are discussed before concluding the paper.

Table 1.: A toy FCA context.

	m_1	m_2	m_3	m_4
g_1	x			x
g_2			x	x
g_3		x		
g_4			x	x

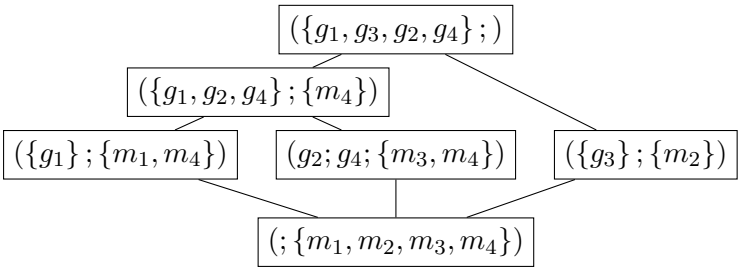


Figure 1.: Concept Lattice for the toy context

2. FCA and pattern structures

2.1. *Formal concept analysis*

FCA is a formalism that can be used for guiding data analysis and knowledge discovery (Ganter and Wille 1999). FCA starts with a formal context and builds a set of formal concepts organized within a concept lattice. A formal context is a triple (G, M, I), where G is a set of objects, M is a set of attributes and I is a relation between G and M, $I \subseteq G \times M$. In Table 1, a cross table for a formal context is shown. A Galois connection between G and M is defined as follows:

$$A' = \{m \in M \mid \forall g \in A, (g, m) \in I\}, \qquad A \subseteq G$$
$$B' = \{g \in A \mid \forall m \in M, (g, m) \in I\}, \qquad B \subseteq M$$

The Galois connection maps a set of objects to the maximal set of attributes shared by all objects and reciprocally. For example, $\{g_1, g_2\}' = \{m_4\}$, while $\{m_4\}' = \{g_1, g_2, g_4\}$, i.e. the set $\{g_1, g_2\}$ is not maximal. Given a set of objects A, we say that A' is the description of A.

Definition 1. *A formal concept is a pair (A, B), where $A \subseteq G$ is a subset of objects, $B \subseteq M$ is a subset of attributes, such that $A' = B$ and $A = B'$, where A is called the extent of the concept, and B is called the intent of the concept.*

A formal concept corresponds to a pair of maximal sets of objects and attributes, i.e. it is not possible to add an object or an attribute to the concept without violating the maximality property. For example a pair $(\{g_1, g_2, g_4\}, \{m_4\})$ is a formal concept. Formal concepts can be partially ordered w.r.t. the extent inclusion (dually, intent inclusion). For example, $(\{g_1\}; \{m_1, m_4\}) \leq (\{g_1, g_2, g_4\}, \{m_4\})$. This partial order of concepts is shown in Figure 1. The number of formal concepts for a given context can be exponential w.r.t. the cardinality of set of objects or set of attributes. It is easy to see that for context (G, G, I_G), where $I_G = \{(x, y) \mid x \in G, y \in G, x \neq y\}$, the number of concepts is equal to $2^{|G|}$.

Table 2.: A toy formal context

	m_1	m_2	m_3	m_4	m_5	m_6
g_1	x					x
g_2		x				x
g_3			x			x
g_4				x		x
g_5					x	

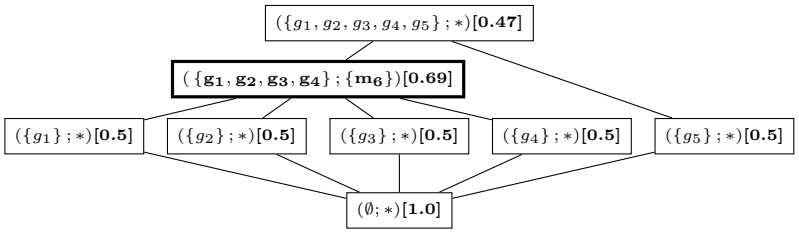


Figure 2.: Concept Lattice for the context in Table 2 with corresponding stability indexes.

2.2. *Stability index of a concept*

The number of concepts in a lattice for real-world tasks can be large. To find the most interesting subset of concepts, different measures can be used such as the stability of the concept (Kuznetsov 2007) or the concept probability and separation (Klimushkin, Obiedkov, and Roth 2010). These measures help extracting the most interesting concepts. However, the last ones are less reliable in noisy data.

Definition 2. *Given a concept c, the concept stability* $\mathrm{Stab}(c)$ *of c is the relative number of subsets of the concept extent (denoted* $\mathrm{Ext}(c)$*), whose description, i.e. the result of $(\cdot)'$, is equal to the concept intent (denoted* $\mathrm{Int}(c)$*).*

$$\mathrm{Stab}(c) := \frac{|\{s \in \wp(\mathrm{Ext}(c)) \mid s' = \mathrm{Int}(c)\}|}{|\wp(\mathrm{Ext}(c))|} \tag{1}$$

Here $\wp(P)$ is the powerset of P. Stability measures how a concept depends on objects in its extent. The larger the stability is the more combinations of objects can be deleted from the context without affecting the intent of the concept, i.e. the intent of the most stable concepts is likely to be a characteristic pattern of a given phenomenon and not an artifact of a dataset. Of course, stable concepts still depend on the dataset, and, consequently some important information can be contained in the unstable concepts. However, the stability can be considered as a good heuristic for selecting concepts because the more stable the concept is the less it depends on the given dataset w.r.t. to object removal.

Example 1. *Figure 2 shows a lattice for the context in Table 2, for simplicity some intents are not given. Extent of the outlined concept c is* $\mathrm{Ext}(c) = \{g_1, g_2, g_3, g_4\}$*, thus, its powerset contains 2^4 elements. Descriptions of 5 subsets of* $\mathrm{Ext}(c)$ *($\{g_1\}, \ldots, \{g_4\}$ and $\emptyset$) are different from* $\mathrm{Int}(c) = \{m_6\}$*, while all other subsets of* $\mathrm{Ext}(c)$ *have a common description equal to $\{m_6\}$. So,* $\mathrm{Stab}(c) = \frac{2^4-5}{2^4} = 0.69$.

One of the fastest algorithm processing a concept lattice L is proposed in (Roth, Obiedkov, and Kourie 2008) with the worst-case complexity of $O(|L|^2)$ where $|L|$ is the size of the concept lattice. The experimental section shows that for a big lattice, the stability computation can take much more time than the construction of the

concept lattice. Thus, the estimation of concept stability is an important question. Here we present an efficient way for such an estimation. It should be noticed that in a lattice the extent of any ancestor of a concept c is a superset of the extent of c, while the extent of any descendant is a subset. Given a concept c and an immediate descendant d, we have $\forall s \subseteq \mathrm{Ext}(d), s'' \subseteq \mathrm{Ext}(d)$, which means that $s' \supseteq \mathrm{Int}(d) \supset \mathrm{Int}(c)$, i.e. $s' \neq \mathrm{Int}(c)$. Thus, we can exclude in the computation of the numerator of stability in (1) all subsets of the extent of a direct descendant c. Thus, the following bound holds:

$$\mathrm{Stab}(c) \leq 1 - \max_{d \in \mathrm{DD}(c)} \frac{1}{2^{\Delta(c,d)}}, \tag{2}$$

where $\mathrm{DD}(c)$ is the set of all direct descendants and $\Delta(c,d)$ is the set-difference between extent of c and extent of d, $\Delta(c,d) = |\mathrm{Ext}(c) \setminus \mathrm{Ext}(d)|$.

Example 2. *With help of (2) we can find all stable concepts (and some unstable), i.e. the concepts with a high stability w.r.t. a threshold θ. If $\theta = 0.97$, we should compute for each concept c in the lattice the following value $md(c) = \min_{d \in \mathrm{DD}(c)} \Delta(c,d)$ and then select concepts verifying $md(c) \geq -\log(1 - 0.97) = 5.06$.*

2.3. *Pattern structures*

Although FCA applies to binary contexts, more complex data such as sequences or graphs can be directly processed as well. For that, pattern structures were introduced in Ganter and Kuznetsov (2001).

Definition 3. *A pattern structure is a triple $(G, (D, \sqcap), \delta)$, where G is a set of objects, $(D, \sqcap)$ is a complete meet-semilattice of descriptions and $\delta : G \to D$ maps an object to a description.*

The lattice operation in the semilattice ($\sqcap$) corresponds to the similarity between two descriptions. Standard FCA can be presented in terms of a pattern structure. In this case, G is the set of objects, the semilattice of descriptions is $(\wp(M), \sqcap)$ and a description is a set of attributes, with the $\sqcap$ operation corresponding to the set intersection ($\wp(M)$ denotes the powerset of M). If $x = \{a, b, c\}$ and $y = \{a, c, d\}$ then $x \sqcap y = x \cap y = \{a, c\}$. The mapping $\delta : G \to \wp(M)$ is given by, $\delta(g) = \{m \in M \mid (g, m) \in I\}$, and returns the description for a given object as a set of attributes.

The Galois connection for a pattern structure $(G, (D, \sqcap), \delta)$ is defined as follows:

$$A^\diamond := \prod_{g \in A} \delta(g), \qquad \text{for } A \subseteq G$$

$$d^\diamond := \{g \in G \mid d \sqsubseteq \delta(g)\}, \qquad \text{for } d \in D$$

The Galois connection makes a correspondence between sets of objects and descriptions. Given a subset of objects A, $A^\diamond$ returns the description which is common to all objects in A. Given a description d, $d^\diamond$ is the set of all objects whose description subsumes d. More precisely, the partial order (or the subsumption order) on D ($\sqsubseteq$) is defined w.r.t. the similarity operation $\sqcap$: $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$, and c is subsumed by d.

Definition 4. *A pattern concept of a pattern structure $(G, (D, \sqcap), \delta)$ is a pair (A, d) where $A \subseteq G$ and $d \in D$ such that $A^\diamond = d$ and $d^\diamond = A$, A is called the concept*

Table 3.: Toy sequential data on patient medical trajectories.

Patient	Trajectory
p^1	$\langle[H_1, \{a\}]; [H_1, \{c, d\}]; [H_1, \{a, b\}]; [H_1, \{d\}]\rangle$
p^2	$\langle[H_2, \{c, d\}]; [H_3, \{b, d\}]; [H_3, \{a, d\}]\rangle$
p^3	$\langle[H_4, \{c, d\}]; [H_4, \{b\}]; [H_4, \{a\}]; [H_4, \{a, d\}]\rangle$

extent and d is called the concept intent.

As in standard FCA, a pattern concept corresponds to the maximal set of objects A whose description subsumes the description d, where d is the maximal common description for objects in A. The set of all concepts can be partially ordered w.r.t. partial order on extents (dually, intent patterns, i.e $\sqsubseteq$), within a concept lattice.

An example of pattern structures is given in Table 3, while the corresponding lattice is depicted in Figure 3.

As stability of concepts only depends on extents, it can be defined by the same procedure for both formal contexts and pattern structures.

3. Sequential pattern structures

Certain phenomena, such as a patient trajectory (clinical history), can be considered as a sequence of events. This section describes how FCA and pattern structures can process sequential data.

3.1. *An example of sequential data*

Imagine that we have medical trajectories of patients, i.e. sequences of hospitalizations, where every hospitalization is described by a hospital name and a set of procedures. An example of sequential data on medical trajectories with three patients is given in Table 3. We have a set of procedures $P = \{a, b, c, d\}$, a set of hospital names $T_H = \{H_1, H_2, H_3, H_4, CL, CH, *\}$, where hospital names are hierarchically organized (by level of generality). H_1 and H_2 are central hospitals (CH), H_3 and H_4 are clinics (CL), and $*$ denotes the root of this hierarchy. The least common ancestor in this hierarchy is denoted by $h_1 \sqcap h_2$, for any $h_1, h_2 \in T_H$, i.e. $H_1 \sqcap H_2 = CH$. Every hospitalization is described by one hospital name and may contain several procedures. The procedure order in each hospitalization is not important in our case. For example, the first hospitalization $[H_2, \{c, d\}]$ for the second patient (p^2) was a stay in hospital H_2 and during this hospitalization the patient underwent procedures c and d. An important task is to find the "characteristic" sequences of procedures and associated hospitals in order to improve hospitalization planning, optimize clinical processes or detect anomalies.

We approach the search for characteristic sequences by finding the most stable concepts in the lattice corresponding to a sequential pattern structure. For the simplification of calculations, subsequences are considered without "gaps", i.e the order of non consequent elements is not taken into account. This is reasonable in this task because experts are interested in regular consecutive events in healthcare trajectories. A sequential pattern structure is a set of sequences and is based on the set of maximal common subsequences (without gaps) between two sequences. Next subsections define partial order on sequences and the corresponding pattern structures.

3.2. *Partial order on complex sequences*

A sequence is constituted of elements from an alphabet. The classical subsequence matching task requires no special properties of the alphabet. Several generalizations of the classical case were made by introducing a subsequence relation based on an itemset alphabet (Agrawal and Srikant 1995) or on a multidimensional and multilevel alphabet (Plantevit et al. 2010). Here, we generalize the previous cases, requiring for an alphabet to form a semilattice $(E, \sqcap_E)$ (We should note that in this paper we consider two semilattices, the first one is related to the characters of the alphabet, $(E, \sqcap_E)$, and the second one is related to pattern structures, $(D, \sqcap)$). Thanks to the formalism of pattern structures we are able to process in a unified way all types of sequential datasets with poset-shaped alphabet (it is mentioned above that any partial order can be transformed into a semilattice). However, some sequential data can have connections between elements, e.g. (Adda et al. 2010), and, thus, cannot be straightforwardly processed by our approach.

Definition 5. *Given a semilattice $(E, \sqcap_E)$, also called an alphabet, a sequence is an ordered list of elements from E. We denote it by $\langle e_1; e_2; \cdots; e_n \rangle$ where $e_i \in E$.*

In this alphabet semilattice $(E, \sqcap_E)$ there is a bottom element $\perp_E$ that can be matched with any other element. Formally, $\forall e \in E, \perp_E = \perp_E \sqcap_E e$. This element is required by the lattice structure, but provides no useful information. Thus, it should be excluded from sequences. The bottom element of E corresponds to the empty set in sequential mining (Agrawal and Srikant 1995), and the empty set is always ignored in this domain.

Definition 6. *A valid sequence $\langle e_1; \cdots; e_n \rangle$ is a sequence where $\forall i \in \{1, \cdots, n\} e_i \neq \perp_E$.*

Definition 7. *Given an alphabet $(E, \sqcap_E)$ and two sequences $t = \langle t_1; ...; t_k \rangle$ and $s = \langle s_1; ...; s_n \rangle$ based on E ($t_q, s_p \in E$), the sequence t is a subsequence of s, denoted $t \leq s$, iff $k \leq n$ and there exist $j_1, ..j_k$ such that $1 \leq j_1 < j_2 < ... < j_k \leq n$ and for all $i \in \{1, 2, ..., k\}$, $t_i \sqsubseteq_E s_{j_i}$, i.e. $t_i \sqcap_E s_{j_i} = t_i$.*

Example 3. *In the running example (Section 3.1), the alphabet is $E = T_H \times \wp(P)$ with the similarity operation $(h_1, P_1) \sqcap (h_2, P_2) = (h_1 \sqcap h_2, P_1 \cap P_2)$, where $h_1, h_2 \in T_H$ are hospitals and $P_1, P_2 \in \wp(P)$ are sets of procedures. Thus, the sequence $ss^1 = \langle [CH, \{c, d\}]; [H_1, \{b\}]; [*, \{d\}] \rangle$ is a subsequence of $p^1 = \langle [H_1, \{a\}]; [H_1, \{c, d\}]; [H_1, \{a, b\}]; [H_1, \{d\}] \rangle$ because if we set $j_i = i + 1$ (Definition 7) then $ss^1_1 \sqsubseteq p^1_{j_1}$ ('CH' is more general than H_1 and $\{c, d\} \subseteq \{c, d\}$), $ss^1_2 \sqsubseteq p^1_{j_2}$ (the same hospital and $\{b\} \subseteq \{b, a\}$) and $ss^1_3 \sqsubseteq p^1_{j_3}$ ('*' is more general than H_1 and $\{d\} \subseteq \{d\}$).*

With complex sequences and this kind of subsequence relation the computation can be hard. Thus, for the sake of simplification, only "contiguous" subsequences are considered, where only the order of consequent elements is taken into account, i.e. given j_1 in Definition 7, $j_i = j_{i-1} + 1$ for all $i \in \{2, 3, ..., k\}$. Since experts are interested in regular consecutive events in healthcare trajectories, such a restriction does make sens for our data. It helps to connect only related hospitalizations.

The next section introduces pattern structures that are based on complex sequences with a general subsequence relation, while the experiments are provided for a "contiguous" subsequence relation.

3.3. *Sequential meet-semilattice*

Based on the previous definitions, we can define the sequential pattern structure used for representing and managing sequences. For that, we make an analogy with the pattern structures for graphs (Kuznetsov 1999) where the meet-semilattice operation $\sqcap$ respects subgraph isomorphism. Thus, we introduce a sequential meet-semilattice respecting subsequence relation. Given an alphabet lattice $(E, \sqcap_E)$, $\mathfrak{S}$ is the set of all valid sequences based on $(E, \sqcap_E)$. $\mathfrak{S}$ is partially ordered w.r.t. Definition 7. $(D, \sqcap)$ is a semilattice on $\mathfrak{S}$, where $D \subseteq \wp(\mathfrak{S})$ such that, if $d \in D$ contains a sequence s, then all subsequences of s should be included into d, $\forall s \in d, \nexists \tilde{s} \leq s : \tilde{s} \notin d$, and the similarity operation is the set intersection for two sets of sequences. Given two patterns $d_1, d_2 \in D$, the set intersection operation ensures that if a sequence s belongs to $d_1 \sqcap d_2$ then any subsequence of s belongs to $d_1 \sqcap d_2$ and thus $d_1 \sqcap d_2 \in D$. As the set intersection operation is idempotent, commutative and associative, $(D, \sqcap)$ is a semilattice.

Example 4. *If pattern $d_1 \in D$ includes sequence $ss^4 = \langle [*, \{c, d\}]; [*, \{b\}] \rangle$ (see Table 4), then it should include also $\langle [*, \{d\}]; [*, \{b\}] \rangle$, $\langle [*, \{c, d\}] \rangle$, $\langle [*, \{d\}] \rangle$ and others. If pattern $d_2 \in D$ includes $ss^{12} = \langle [*, \{a\}]; [*, \{d\}] \rangle$, then it should include $\langle [*, \{a\}] \rangle$, $\langle [*, \{d\}] \rangle$ and $\langle \rangle$. Thus the intersection of two sets d_1 and d_2 is equal to the set $\{\langle [*, \{d\}] \rangle, \langle \rangle\}$.*

The next proposition stems from the aforementioned and will be used in the proofs in the next section.

Proposition 1. *Given $(G, (D, \sqcap), \delta)$ and $x, y \in D$, $x \sqsubseteq y$ if and only if $\forall s^x \in x$ there is a sequence $s^y \in y$, such that $s^x \leq s^y$.*

The set of all possible subsequences for a given sequence can be large. Thus, it is more efficient to consider a pattern $d \in D$ as a set of only maximal sequences $\tilde{d}$, $\tilde{d} = \{s \in d \mid \nexists s^* \in d : s^* \geq s\}$. Furthermore, every pattern will be given only by the set of all maximal sequences. For example, $\{p^2\} \sqcap \{p^3\} = \{ss^6, ss^7, ss^8\}$ (see Tables 3 and 4), i.e. $\{ss^6, ss^7, ss^8\}$ is the set of all maximal sequences specifying the intersection of p^2 and p^3. Similarly we have $\{ss^6, ss^7, ss^8\} \sqcap \{p^1\} = \{ss^4, ss^5\}$. Note that representing a pattern by the set of all maximal sequences allows for an efficient implementation of the intersection "$\sqcap$" of two patterns (in Section 5.1 we give more details on similarity operation w.r.t. a contiguous subsequence relation).

Example 5. *The sequential pattern structure for our example (Subsection 3.1) is $(G, (D, \sqcap), \delta)$, where $G = \{p^1, p^2, p^3\}$, $(D, \sqcap)$ is the semilattice of sequential descriptions, and δ is the mapping associating an object in G to a description in D shown in Table 3. Figure 3 shows the resulting lattice of sequential pattern concepts for this particular pattern structure $(G, (D, \sqcap), \delta)$.*

4. Projections of sequential pattern structures

Pattern structures are hard to process due to the large number of concepts in the concept lattice, the complexity of the involved descriptions and the similarity operation. Moreover, a given pattern structure can produce a lattice with a lot of patterns which are not interesting for an expert. *Can we save computational time by avoiding to compute "useless" patterns?* Projections of pattern structures "simplify" to some degree the computation and allow one to work with a reduced description. In fact, projections can be considered as filters on patterns respecting mathematical proper-

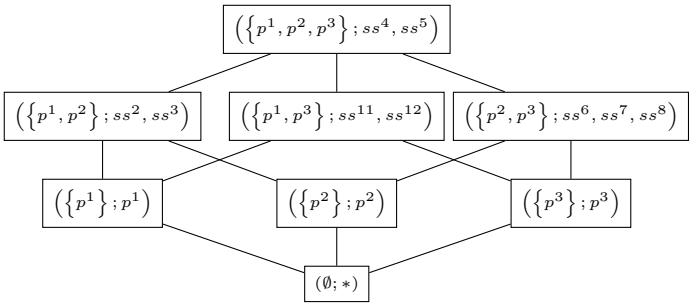


Figure 3.: The concept lattice for the pattern structure given by Table 3. Concept intents reference to sequences in Tables 3 and 4.

Table 4.: Subsequences of patient sequences in Table 3.

	Subsequences
ss^1	$\langle[CH, \{c, d\}]; [H_1, \{b\}]; [*, \{d\}]\rangle$
ss^2	$\langle[CH, \{c, d\}]; [*, \{b\}]; [*, \{d\}]\rangle$
ss^3	$\langle[CH, \{\}]; [*, \{d\}]; [*, \{a\}]\rangle$
ss^4	$\langle[*, \{c, d\}]; [*, \{b\}]\rangle$
ss^5	$\langle[*, \{a\}]\rangle$
ss^6	$\langle[*, \{c, d\}]; [CL, \{b\}]; [CL, \{a\}]\rangle$
ss^7	$\langle[CL, \{d\}]; [CL, \{\}]\rangle$
ss^8	$\langle[CL, \{\}]; [CL, \{a, d\}]\rangle$
ss^9	$\langle[CH, \{c, d\}]\rangle$
ss^{10}	$\langle[CL, \{b\}]; [CL, \{a\}]\rangle$
ss^{11}	$\langle[*, \{c, d\}]; [*, \{b\}]\rangle$
ss^{12}	$\langle[*, \{a\}]; [*, \{d\}]\rangle$

ties. These properties ensure that the projection of a semilattice is a semilattice and that projected concepts are related to original ones (Ganter and Kuznetsov 2001). Moreover, the stability measure of projected concepts never decreases w.r.t the original concepts. We introduce projections on sequential patterns revising Ganter and Kuznetsov (2001). It is necessary to provide an extended definition of projection in order to deal with interesting projections for real-world sequential datasets.

Definition 8 (Ganter and Kuznetsov (2001))**.** *A projection $\psi : D \to D$ is an interior operator, i.e. it is (1) monotone ($x \sqsubseteq y \Rightarrow \psi(x) \sqsubseteq \psi(y)$), (2) contractive ($\psi(x) \sqsubseteq x$) and (3) idempotent ($\psi(\psi(x)) = \psi(x)$).*

Definition 9. *A projected pattern structure $\psi((G, (D, \sqcap), \delta))$ is a pattern structure $(G, (D_\psi, \sqcap_\psi), \psi \circ \delta)$, where $D_\psi = \psi(D) = \{d \in D \mid \exists d^* \in D : \psi(d^*) = d\}$ and $\forall x, y \in D, x \sqcap_\psi y := \psi(x \sqcap y)$.*

Note that in (Ganter and Kuznetsov 2001) $\psi((G, (D, \sqcap), \delta)) = (G, (D, \sqcap), \psi \circ \delta)$. Our definition allows one to use a wider set of projections. In fact all projections that we describe for sequential pattern structures below require Definition 9. Now we should show that $(D_\psi, \sqcap_\psi)$ is a semilattice.

Proposition 2. *Given a semilattice $(D, \sqcap)$ and a projection ψ, for all $x, y \in D$ $\psi(x \sqcap y) = \psi(\psi(x) \sqcap y)$.*

Proof. (1) $\psi(x) \sqsubseteq x$, thus, $x, y \sqsupseteq (x \sqcap y) \sqsupseteq (\psi(x) \sqcap y) \sqsupseteq \psi(\psi(x) \sqcap y)$
(2) $x \sqsubseteq y \Rightarrow \psi(x) \sqsubseteq \psi(y)$, thus, $\psi(x \sqcap y) \sqsupseteq \psi(\psi(x) \sqcap y)$
(3) $\psi(x \sqcap y) \sqcap \psi(x) \sqcap y \underset{\psi(x \sqcap y) \sqsubseteq \psi(x)}{=} \psi(x \sqcap y) \sqcap y \underset{\psi(x \sqcap y) \sqsubseteq y}{=} \psi(x \sqcap y)$,
then $(\psi(x) \sqcap y) \sqsupseteq \psi(x \sqcap y)$ and $\psi(\psi(x) \sqcap y) \sqsupseteq \psi(\psi(x \sqcap y)) = \psi(x \sqcap y)$
(4) From (2) and (3) it follows that $\psi(x \sqcap y) = \psi(\psi(x) \sqcap y)$.

□

Corollary 1. $X_1 \sqcap_\psi X_2 \sqcap_\psi \cdots \sqcap_\psi X_N = \psi(X_1 \sqcap X_2 \sqcap \cdots \sqcap X_N)$

Proof. It can be prooven by induction.

(1) $X_1 \sqcap_\psi X_2 = \psi(X_1 \sqcap X_2)$ by Definition 9.
(2) If $X_1 \sqcap_\psi \cdots \sqcap_\psi X_K = \psi(X_1 \sqcap \cdots \sqcap X_K)$, then
$$X_1 \sqcap_\psi \cdots \sqcap_\psi X_K \sqcap_\psi X_{K+1} = \psi(X_1 \sqcap \cdots \sqcap X_K) \sqcap_\psi X_{K+1} =$$
$$= \psi(\psi(X_1 \sqcap \cdots \sqcap X_K) \sqcap X_{K+1}) \underset{\text{Proposition 2}}{=} \psi(X_1 \sqcap \cdots \sqcap X_{K+1})$$

□

Corollary 2. *Given a semilattice $(D, \sqcap)$ and a projection ψ, $(D_\psi, \sqcap_\psi)$ is a semilattice, i.e. $\sqcap_\psi$ is commutative, associative and idempotent.*

The concepts of a pattern structure and a projected pattern structure are connected through Proposition 3. This proposition can be found in Ganter and Kuznetsov (2001), but thanks to Corollary 1, it is valid in our case.

Proposition 3. *Given a concept (A, d) in $\psi((G, (D, \sqcap), \delta))$, the extent A is an extent in $(G, (D, \sqcap), \delta)$. Given a concept (A, d_ψ) in $\psi((G, (D, \sqcap), \delta))$, the intent d_ψ is of the form $d_\psi = \psi(d)$, where (A, d) is a concept in $(G, (D, \sqcap), \delta)$.*

Moreover, while preserving the extents of some concepts, projections cannot decrease the stability of the projected concepts, i.e. if the projection preserves a stable concept, then its stability (Definition 2) can only increase.

Proposition 4. *Given a pattern structure $(G, (D, \sqcap), \delta)$, its concept c and a projected pattern structure $(G, (D_\psi, \sqcap_\psi), \psi \circ \delta)$, and the projected concept $\tilde{c}$, if the concept extents are equal ($\mathrm{Ext}(c) = \mathrm{Ext}(\tilde{c})$) then $\mathrm{Stab}(c) \leq \mathrm{Stab}(\tilde{c})$.*

Proof. Concepts c and $\tilde{c}$ have the same extent. Thus, according to Definition 2, in order to prove the proposition, it is enough to prove that for any subset $A \subseteq \mathrm{Ext}(c)$, if $A^\diamond = \mathrm{Int}(c)$ in the original pattern structure, then $A^\diamond = \mathrm{Int}(\tilde{c})$ in the projected one.

Suppose that $\exists A \subset \mathrm{Ext}(c)$ such that $A^\diamond = \mathrm{Int}(c)$ in the original pattern structure and $A^\diamond \neq \mathrm{Int}(\tilde{c})$ in the projected one. Then there is a descendant concept $\tilde{d}$ of $\tilde{c}$ in the projected pattern structure such that $A^\diamond = \mathrm{Int}(\tilde{d})$ in the projected lattice. Then there is an original concept d for the projected concept $\tilde{d}$ with the same extent $\mathrm{Ext}(d)$. Then $A^\diamond \sqsupseteq \mathrm{Int}(d) \sqsupset \mathrm{Int}(c)$ and, so, $A^\diamond$ cannot be equal to $\mathrm{Int}(c)$ in the original lattice. Contradiction. □

Now we are going to present two projections of sequential pattern structures. The first projection comes from the following observation. In many cases it may be more interesting to analyze quite long subsequences rather than short ones. This kind of projections is called *Minimal Length Projection* (MLP) and it depends on the minimal length parameter ℓ for the sequences in a pattern. The corresponding function ψ maps a pattern without short sequences to itself, and a sequence with short sequences to the pattern containing only long sequences w.r.t. a given length threshold. Later, propositions 1 and 5 state that MLP is coherent with Definition 8.

Definition 10. *The function $\psi_{MLP} : D \to D$ of minimal length ℓ is defined as*

$$\psi_{MLP}(d) = \{s \in d \mid length(s) \geq \ell\}$$

Example 6. *If we prefer common subsequences of length $\ell \geq 3$, then between p^2 and p^3 in Table 3 there is only one maximal common subsequence, ss^6 in Table 4, while ss^7 and ss^8 are too short to be considered. Figure 4a shows the lattice of the projected pattern structure (Table 3) with patterns of length greater or equal to* 3.

Proposition 5. *The function ψ_{MLP} is a monotone, contractive and idempotent function on the semilattice $(D, \sqcap)$.*

Proof. The contractivity and idempotency are quite clear from the definition. It remains to prove the monotonicity.

If $X \sqsubseteq Y$, where X and Y are sets of sequences, then for every sequence $x \in X$ there is a sequence $y \in Y$ such that $x \leq y$ (Proposition 1). We should show that $\psi(X) \sqsubseteq \psi(Y)$, or in other words for every sequence $x \in \psi(X)$ there is a sequence $y \in \psi(Y)$, such that $x \leq y$. Given $x \in \psi(X)$, since $\psi(X)$ is a subset of X and $X \sqsubseteq Y$, there is a sequence $y \in Y$ such that $x \leq y$, with $|y| \geq |x| \geq \ell$ (ℓ is a parameter of MLP), and thus, $y \in \psi(Y)$. □

Another important type of projections is related to a variation of the lattice alphabet $(E, \sqcap_E)$. One possible variation of the alphabet is to ignore certain fields in the elements. For example, if a hospitalization is described by a hospital name and a set of procedures, then either hospital or procedures can be ignored in similarity computation. For that, in any element the set of procedures should be substituted by $\emptyset$, or the hospital by $*$ ("arbitrary hospital") which is the most general element of the taxonomy of hospitals.

Another variation of the alphabet is to require that some field(s) should not be empty. For example, we want to find patterns with non-empty set of procedures or the element $*$ of the hospital taxonomy is not allowed in elements of a sequence. Such variations are easy to realize within our approach. For this, when computing the similarity operation between elements of the alphabet, one should check if the result contains empty fields and, if yes, should substitute the result by $\bot$. This variation is useful, as it is shown in the experimental section, but is rather difficult to define within more classical frequent sequence mining approaches, which will be discussed later.

Example 7. *An expert is interested in finding sequential patterns describing how a patient changes hospitals, but with little interest in procedures. Thus, any element of the alphabet lattice, containing a hospital and a non-empty set of procedures can be projected to an element with the same hospital, but with an empty set of procedures.*

Example 8. *An expert is interested in finding sequential patterns containing some information about the hospital in every hospitalization, and the corresponding procedures, i.e. hospital field in the patterns cannot be equal to $*$, e.g., ss^5 is an invalid pattern, while ss^6 is a valid pattern in Table 4. Thus, any element of the alphabet semilattice with $*$ in the hospital field can be projected to the $\bot_E$. Figure 4b shows the lattice corresponding to the projected pattern structure (Table 3) defined by a projection of the alphabet semilattice.*

Below we formally define how the alphabet projection of a sequential pattern structure should be processed. Intuitively, every sequence in a pattern should be substituted with another sequence, by applying the alphabet projection to all its elements. However, the result can be an incorrect sequence, because $\bot_E$ cannot belong to a valid sequence. Thus, sequences in a pattern should be "developed" w.r.t. $\bot_E$, as it is explained below.

Definition 11. *Given an alphabet $(E, \sqcap_E)$, a projection of the alphabet ψ and a sequence $s = \langle s_1, \cdots, s_n \rangle$ based on E, the projection $\psi(s)$ is the sequence $\tilde{s} = \langle \tilde{s}_1, \cdots, \tilde{s}_n \rangle$, such that $\tilde{s}_i = \psi(s_i)$.*

Here, it should be noticed that $\tilde{s}$ is not necessarily a valid sequence (see Definition 6), since it can include $\perp_E$ as an element. However, in sequential pattern structures, elements should include only valid sequences (see Section 3.3).

Definition 12. *Given an alphabet $(E, \sqcap_E)$, a projection of the alphabet ψ_E, an alphabet projection for the sequential pattern structure $\psi(d)$ is the set of valid sequences smaller than the projected sequences from d:*

$$\psi(d) = \{s \in \mathfrak{S} | (\exists t \in d) s \leq \psi_E(t)\},$$

where $\mathfrak{S}$ is the set of all valid sequences based on $(E, \sqcap_E)$.

Example 9. *$\{ss^6\} = \{\langle [*, \{c, d\}]; [CL, \{b\}]; [CL, \{a\}] \rangle\}$ is an alphabet-projected pattern for the pattern $\{ss^{10}\} = \{\langle [CL, \{b\}]; [CL, \{a\}] \rangle\}$, where the alphabet lattice projection is given in Example 8.*

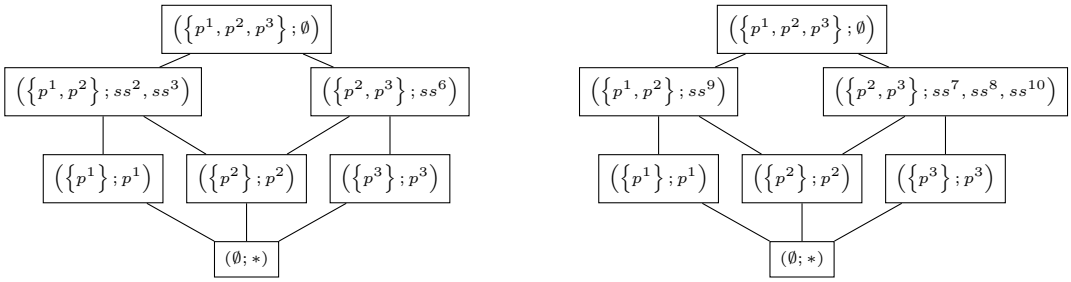
In the case of contiguous subsequences, $\{\langle [CH, \{c, d\}] \rangle\}$ is an alphabet-projected pattern for the pattern $\{ss^2\} = \{\langle [CH, \{c, d\}]; [*, \{b\}]; [*, \{d\}] \rangle\}$, where the alphabet lattice projection is given by projecting every element with medical procedure b to the element with the same hospital and with the same set of procedures excluding b. The projection of sequence ss^2 is $\langle [CH, \{c, d\}]; [*, \{\}]; [*, \{d\}] \rangle$, but $[*, \{\}] = \perp_E$, and, thus, in order to project the pattern $\{ss^2\}$ the projected sequence is substituted by its maximal subsequences, i.e. $\psi(\{\langle [CH, \{c, d\}]; [*, \{b\}]; [*, \{d\}] \rangle\}) = \{\langle [CH, \{c, d\}] \rangle\}$.

Proposition 6. *Considering an alphabet $(E, \sqcap_E)$, a projection of the alphabet ψ, a sequential pattern structure $(G, (D, \sqcap), \delta)$, the alphabet projection (see Definition 12) is monotone, contractive and idempotent.*

Proof. This projection is idempotent, since the projection of the alphabet is idempotent and only the projection of the alphabet can change the elements appearing in sequences.

It is contractive because for any pattern $d \in D$ and any sequences $s \in d$, a projection of the sequence $\tilde{s} = \psi(s)$ is a subsequence of s. In Definition 12 the projected sequences should be substituted by their subsequences in order to avoid $\perp_E$, building the sets $\{\tilde{s}^i\}$. Thus, s is a supersequence for any $\tilde{s}^i$, and, so, the projected pattern $\tilde{d} = \psi(d)$ is subsumed by the pattern d.

Finally, we should show monotonicity. Given two patterns $x, y \in D$, such that $x \sqsubseteq y$, i.e. $\forall s^x \in x, \exists s^y \in y : s^x \leq s^y$, consider the projected sequence of s^x, $\psi(s^x)$. As $s^x \leq s^y$ for some s^y then for some $j_0 < \cdots < j_{|s^x|}$ (see Definition 7) $s^x_i \sqsubseteq_E s^y_{j_i}$ $(i \in 1, 2, ..., |s^x|)$, then $\psi(s^x_i) \sqsubseteq_E \psi(s^y_{j_i})$ (by the monotonicity of the alphabet projection), i.e. the projected sequence preserves the subsequence relation. Thus, the set of allowed subsequences of s^x is a subset of the set of allowed subsequences of s^y. Hence, the alphabet projection of the pattern preserves pattern subsumption relation, $\psi(x) \leq \psi(y)$ (Proposition 1), i.e. the alphabet projection is monotone. □



(a) MLP projection, $l = 3$

(b) Projection removing '*' in the hospital field

Figure 4.: The projected concept lattices for the pattern structure given by Table 3. Concept intents refer to the sequences in Tables 3 and 4.

5. Sequential pattern structure evaluation

5.1. *Implementation*

Nearly any state-of-the-art FCA algorithm can be adapted to process pattern structures. We adapted the `AddIntent` algorithm (Merwe, Obiedkov, and Kourie 2004), as the lattice structure is important for us to calculate stability (see an algorithm for calculating stability in (Roth, Obiedkov, and Kourie 2008)). To adapt the algorithm to our needs, every set intersection operation on attributes is substituted with the semilattice operation $\sqcap$ on corresponding patterns, while every subset checking operation is substituted with the semilattice order checking $\sqsubseteq$, in particular all $(\cdot)'$ are substituted with $(\cdot)^\diamond$.

The next question is how the semilattice operation $\sqcap$ and subsumption relation $\sqsubseteq$ can be implemented for contiguous sequences. Given two sets of sequences $S = \{s^1, ...s^n\}$ and $T = \{t^1, ..., t^m\}$, the similarity of these sets $S \sqcap T$, is calculated according to Section 3.3, i.e. maximal sequences among all common subsequences for any pair of sequences s^i and t^j.

To find all common subsequences of two sequences, the following observations can be useful. If $ss = \langle ss_1; ...; ss_l\rangle$ is a subsequence of $s = \langle s_1; ...; s_n\rangle$ with $j_i^s = k^s + i$, i.e. $ss_i \sqsubseteq_E s_{k^s+i}$ (Definition 7: k^s is the index difference from which ss is a contiguous subsequence of s) and a subsequence of $t = \langle t_1; ...; t_m\rangle$ with $j_i^t = k^t + i$, i.e. $ss_i \sqsubseteq_E t_{k^t+i}$, then for any index $i \in \{1, 2, ..., l\}$, $ss_i \sqsubseteq_E (s_{j_i^s} \sqcap t_{j_i^t})$. Thus, to find all maximal common subsequences of s and t, we first align s and t in all possible ways. For each alignment of s and t we compute the resulting intersection. Finally, we keep only the maximal intersected subsequences.

For example, let us consider two possible alignments of s^1 and s^2:

$$\begin{array}{rlrl} s^1 = & \langle\{a\};\{c,d\};\{b,a\};\ \{d\}\ \rangle & s^1 = & \langle\{a\};\{c,d\};\{b,a\};\ \{d\}\ \rangle \\ s^2 = & \langle\{c,d\};\{b,d\};\{a,d\}\rangle & s^2 = & \langle\{c,d\};\{b,d\};\{a,d\}\rangle \\ ss^l = & \langle\ \emptyset\ ;\{d\}\ \rangle & ss^r = & \langle\{c,d\};\ \{b\}\ ;\ \{d\}\ \rangle \end{array}$$

The left intersection ss^l is not retained, as it is not maximal ($ss^l < ss^r$), while the right intersection ss^r is kept.

The complexity of the alignment for two sequences s and t is $O(|s| \cdot |t| \cdot \gamma)$, where γ is the complexity of computing a common ancestor in the alphabet lattice $(E, \sqcap)$.

5.2. *Experiments and discussion*

The experiments are carried out on a MacBook Pro with a 2.5GHz Intel Core i5, 8GB of RAM Memory running OS X 10.6.8. The algorithms are not parallelized

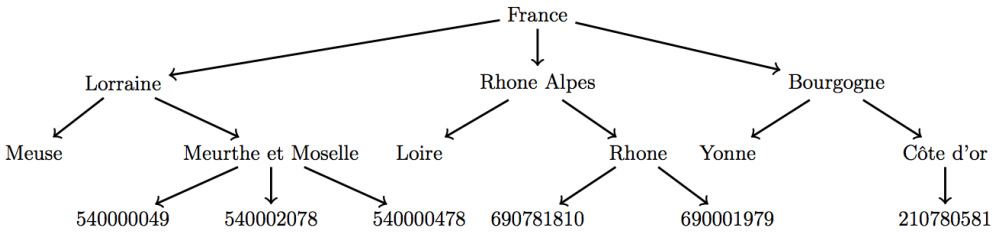


Figure 5.: A geographical taxonomy of the healthcare institution

and are coded in C++.

Our use-case dataset comes from a French healthcare system, called PMSI[2] (Fetter et al. 1980). Each element of a sequence has a "complex" nature. The dataset contains 500 patients suffering from *lung cancer*, who live in the Lorraine region (Eastern France). Every patient is described as a sequence of hospitalizations without any time-stamp. A hospitalization is a tuple with three elements: (i) healthcare institution (e.g. university hospital of Nancy (CHU_{Nancy})), (ii) reason for the hospitalization (e.g. a cancer disease), and (iii) set of medical procedures that the patient undergoes. An example of a medical trajectory is given below:

$$\langle [\text{CHU}_{Nancy}, \text{Cancer}, \{mp_1, mp_2\}] \; ; \; [\text{CH}_{Paris}, \text{Chemo}, \{\}] \quad ; [\text{CH}_{Paris}, \text{Chemo}, \{\}] \rangle \, .$$

This sequence represents a patient trajectory with three hospitalizations. It expresses that the patient was first admitted to the university hospital of Nancy (CHU_{Nancy}) for a cancer problem as a reason, and underwent procedures mp_1 and mp_2. Then he had two consequent hospitalizations in the general hospital of Paris (CH_{Paris}) for chemotherapy with no additional procedure. Substituting the same consequent hospitalizations by the number of repetitions, we have a shorter and more understandable trajectory. For example, the above pattern is transformed into two hospitalizations where the first hospitalization repeats once and the second twice:

$$\langle [\text{CHU}_{Nancy}, \text{Cancer}, \{mp_1, mp_2\}] \times [1]; [\text{CH}_{Paris}, \text{Chemo}, \{\}] \times [2] \rangle \, .$$

Diagnoses are coded according to the 10^{th} International Classification of Diseases (ICD10). Based on this coding, diagnoses could be described at 5 levels of granularity: root, chapter, block, 3-character, 4-character, terminal nodes. This taxonomy has 1544 nodes. The healthcare institution is associated with a geographical taxonomy of 4 levels, where the first level refers to the root (France) and the second, the third and the fourth levels correspond to administrative region, administrative department and hospital respectively. Figure 5 presents University Hospital of Nancy (code: 540002078) as a hospital in Meurthe et Moselle, which is a department in Lorraine, region of France. This taxonomy has 304 nodes. The *medical procedures* are coded according to the French nomenclature "Classification Commune des Actes Médicaux (CCAM)". The distribution of sequence lengths is shown in Figure 6.

With 500 patient trajectories, the computation of the whole lattice is infeasible. We are not interested in all possible frequent trajectories, but rather in trajectories which answer medical analysis questions. An expert may know the minimal size of

[2]Programme de Médicalisation des Sytèmes d'Information.

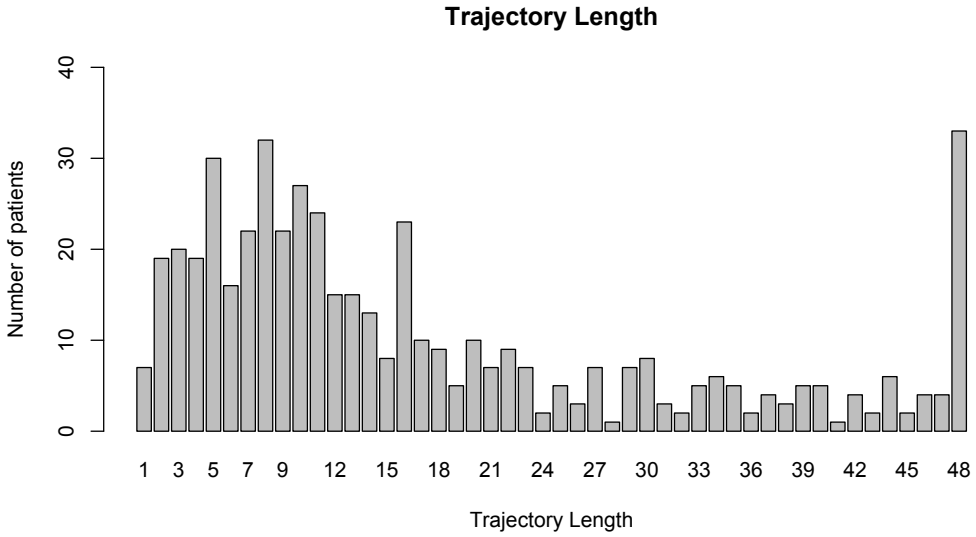
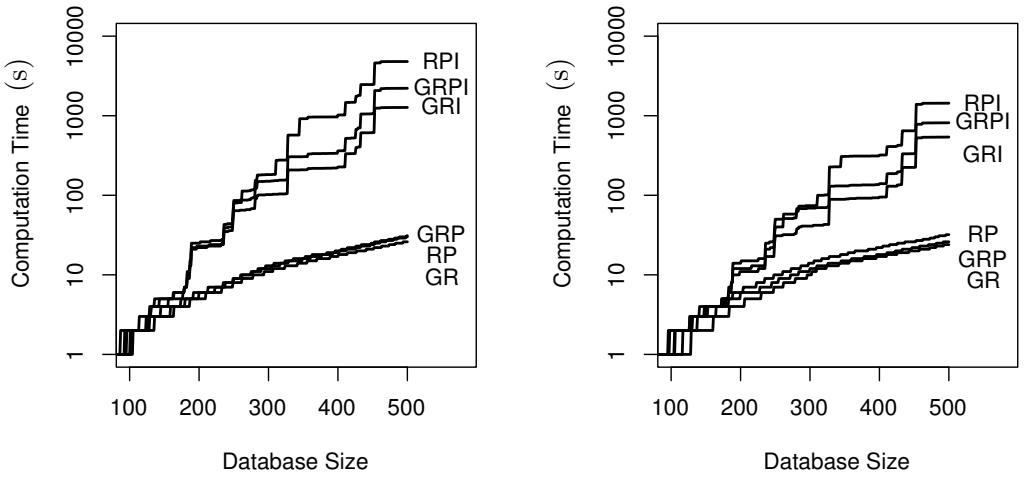


Figure 6.: The length distribution of sequences in the dataset

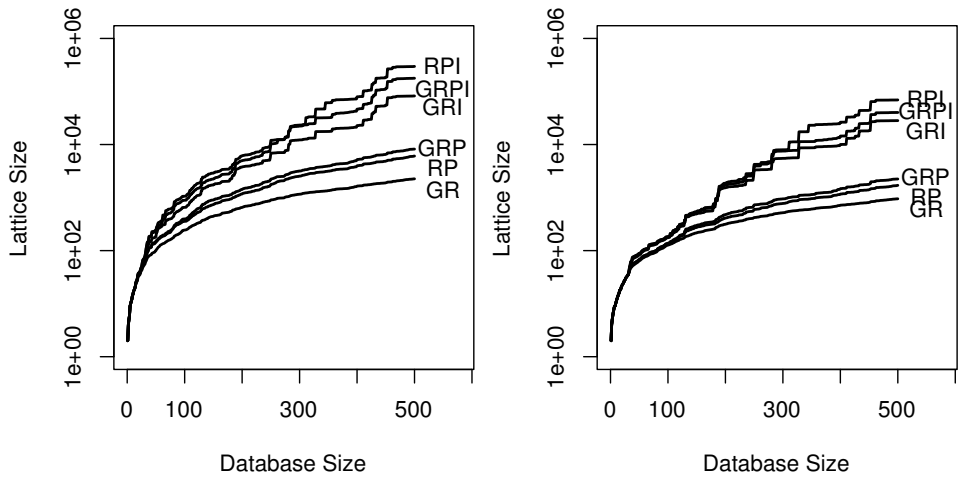


(a) MLP projection, $\ell = 2$ (b) MLP projection, $\ell = 3$

Figure 7.: Computational time for different projections

trajectories that he is interested in, i.e. setting the MLP projection. We use the MLP projection of length 2 and 3 and take into account that most of the patients has at least 2 hospitalizations in the trajectory (see Figure 6).

Figure 7 shows computational times for different projections as a function of dataset size. Figure 7a shows different alphabet projections for MLP projection with $\ell = 2$, while Figure 7b for MLP with $\ell = 3$. Every alphabet projection is given by the name of fields, that are considered within the projection: `G` corresponds to hospital geo-location, `R` is the reason for a hospitalization, `P` is medical procedures and `I` is repetition interval, i.e. the number of consequent hospitalizations with the same reason. We can see from these figures that MLP allows one to save some computational resources with increasing of ℓ. The difference in computational time between $\ell = 2$ and $\ell = 3$ projections is significant, especially for time consuming cases. Even a bigger variation can be noticed for the alphabet projections. For example, computation of the `RPI` projection takes 100 times more resources than



(a) MLP projection, $\ell = 2$ (b) MLP projection, $\ell = 3$

Figure 8.: Lattice size for different projections

Table 5.: Interesting concepts, for different projections.

#	Projection	Intent	Stab. Rank	Support
1	GR	$\langle[Lorraine, C341\ Lung\ Cancer]\rangle$	1	287
2	$GR2$	$\langle[Lorraine, Respiratory\ Disease]; [CHU_{Nancy}, Lung\ Cancer]\rangle$	26	22
3	$GR3$	$\langle[Lorraine, Chemotherapy] \times 4\rangle$	1	176
4	$RPI3$	$\langle[Preparation\ for\ Chemotherapy, \{Lung\ \ Radiography\}];\ [Chemotherapy] \times [3, 4]\rangle$	5	36

any from `GRP, RP, GR, GRP`.

The same dependency can be seen in Figure 8, where the number of concepts for every projection is shown. Consequently, it is important for an expert to provide a strict projection that allows him to answer his questions in order to save computational time and memory.

Table 5 shows some interesting concept intents with the corresponding support and ranking w.r.t. concept stability. For example the concept #1 is obtained under the projection GR (i.e., we consider only hospital and reason), with the intent $\langle[Lorraine, C341\ Lung\ Cancer]\rangle$, where `C341 Lung Cancer` is a special kind of lung cancer (malignant neoplasm in Upper lobe, bronchus or lung). This concept is the most stable concept in the lattice for the given projection, and the size of the concept extent is 287 patients.

One of the questions that the analyst would like to address here is *"Where do patients stay (i.e. hospital location) during their treatment, and for which reason ?"*. To answer this question, we consider only healthcare institutions and reason fields, requiring both to "hold" some information and we use the MLP projection of length 2 and 3 (i.e. projections $GR2$ and $GR3$). Nearly all frequent trajectories show that patients usually are treated in the same region. However, *pattern #2* obtained under $GR2$ projection shows that, *"22 patients were first admitted in some healthcare institution in Lorraine region for a problem related to the respiratory system and then they were treated for a lung cancer in University Hospital of Nancy."*

Another interesting question is *"What are the sequential relations between hospitalization reasons and the corresponding procedures?"*. To answer this question, we are not interested in healthcare institutions. Thus, any alphabet element is projected by substituting healthcare institution field with '*'. As hospitalization reason is important in each hospitalization, any alphabet element without the hospitalization reason is of no use and is projected to the bottom element $\perp_E$ of the alphabet.

Such projections are called $RPI2$ or $RPI3$, meaning that we consider the fields "Reason" and "Procedures", while the reason should not be empty and the MLP parameter is 2 or 3. *Pattern #4* trivially states that, *"36 patients with lung cancer are hospitalized once for the preparation of chemotherapy and during this hospitalization they undergo lung radiography. Afterwards, they are hospitalized between 3 and 4 times for chemotherapy."*

Variability is high in healthcare processes and affects many aspects of healthcare trajectories: patients, medical habits and protocols, healthcare organisation, availability of treatments and settings... Mining sequential pattern structures is an interesting approach for finding regularities across one or several dimensions of medical trajectories in a population of patients. It is flexible enough to help healthcare managers to answer specific questions regarding the natural organisation of care processes and to further compare them with expected or desirable processes. The use of taxonomies plays also a key role in finding the right level of description of sequential patterns and reducing the interpretation overhead.

6. Related work

Agrawal and Srikant (1995) introduced the problem of mining sequential patterns over large sequential databases. Formally, given a set of sequences, where each sequence is a list of transactions ordered by time and each transaction is a set of items, the problem amounts to find all frequent subsequences that appear a sufficient number of times with a user-specified minimum support threshold (*minsup*). Following the work of Agrawal and Srikant many studies have contributed to the efficient mining of sequential patterns (Mooney and Roddick 2013). Most of them are based on the antimonotonicity property (used in *Apriori*), which states that any super pattern of a non-frequent pattern cannot be frequent. The main algorithms are PrefixSpan (Pei et al. 2001a), SPADE (Zaki 2001), SPAM (Ayres et al. 2002), PSP (Masseglia, Cathala, and Poncelet 1998), DISC (Chiu, Wu, and Chen 2004), PAID (Yang, Kitsuregawa, and Wang 2006) and FAST (Salvemini et al. 2011). All these algorithms aim at discovering sequential patterns from a set of sequences of itemsets such as customers who frequently buy DVDs of episodes I, II and III of Stars Wars, then buy within 6 months episodes IV, V, VI of the same famous epic space opera.

Many studies about sequential pattern discovery focus on single-dimensional sequences. However, in many situations, the database is multidimensional in the sense that items can be of different nature. For example, a consumer database can hold information such as article price, gender of the customer, location of the store and so on. Pinto et al. (2001) proposed the first work for mining multidimensional sequential patterns. In this work, a *multidimensional sequential database* is defined as a schema $(ID, D_1, ..., D_m, S)$, where ID is a unique customer identifier, $D_1, ..., D_m$ are dimensions describing the data and S is the sequence of itemsets. A *multidimensional sequence* is defined as a vector $\langle\{d_1, d_2, ..., d_m\}, S_1, S_2, ..., S_l\rangle$ where $d_i \in D_i$ for $(i \leqslant m)$ and $S_1, S_2, ..., S_l$, are the itemsets of sequence S. For instance, $\langle\{Metz, Male\}, \{mp_1, mp_2\}, \{mp_3\}\rangle$ describes a male patient who underwent procedures mp_1 and mp_2 in Metz and then underwent mp_3 also in Metz. Here, dimensions remain constant over time, such as the location of the treatment. This means that it is not possible to have a pattern indicating that when the patient underwent procedures mp_1 and mp_2 in Metz then he underwent mp_3 in Nancy. Among other proposals, Yu and Chen (2005) proposed two methods AprioriMD

and PrefixMDSpan for mining multidimensional sequential patterns in the web domain. This study considers pages, sessions and days as dimensions. Actually, these three different dimensions can be projected into a single dimension corresponding to web pages, gathering web pages visited during a same session and ordering sessions w.r.t the day as order.

In real world applications, each dimension can be represented at different levels of granularity, by using a poset. For example, apples in a market basket analysis can be either described as fruits, fresh food or food. The interest lies in the capacity of extracting more or less general/specific multidimensional sequential patterns and overcome problems of excessive granularity and low support. Srikant and Agrawal (1996) proposed GSP which uses posets for extracting sequential patterns. The basic approach is based on replacing every item with all the ancestors in the poset and then the frequent sequences are generated. This approach is not scalable in a multidimensional context because the size of the database becomes the product of maximum height of the posets and number of dimensions.

Plantevit et al. (2010) defined a *multidimensional sequence* as an ordered list of multidimensional items, where a *multidimensional item* is a tuple $(d_1, ..., d_m)$ and d_i is an item associated with the i^{th} dimension. They proposed M^3SP, an approach taking both aspects into account where each dimension is represented at different levels of granularity, by using a poset. M^3SP is able to search for sequential patterns with the most appropriate level of granularity. Their approach is based on the extraction of the most specific frequent multidimensional items, which are then used as alphabet to rephrase the original database. Then, M^3SP uses a standard sequential pattern mining algorithm to extract multidimensional sequential patterns. However, M^3SP is not adapted to mine sequential databases, where sequences are defined over a combination of sets of items and items lying in a poset. Then it is not possible to have a pattern indicating that when the patient went to uh_p for a problem of cancer ca, where he underwent procedures mp_1 and mp_2, then he went to gh_l for the same medical problem ca, where he underwent mp_3 (i.e, $\langle(uh_p, ca, \{mp_1, mp_2\}), (gh_l, ca, \{mp_3\})\rangle$). Our approach allows us to process such kind of patterns and in addition the elements of sequences are even more general. For example, beside multidimensional and multilevel sequences, sequences of graphs fall under our definition. Moreover, frequent subsequence mining gives rise to a lot of subsequences which can be hardly analyzed by an expert. Since our approach is based on Formal Concept Analysis (FCA) (Ganter and Wille 1999), we can use efficient relevance indexes defined in FCA.

This paper is not the first attempt to use FCA for the analysis of sequential data. Ferré (2007) processes sequential datasets based on a "simple" alphabet without involving any partial order. In Casas-Garriga (2005) only sequences of itemsets are considered. All closed subsequences are firstly mined and then regrouped by a specialized algorithm in order to obtain a lattice similar to the FCA lattice. This approach was not verified experimentally. Moreover, compared with both approaches, i.e. Ferré (2007) and Casas-Garriga (2005), our approach suggests a more general definition of sequences and, thanks to pattern structures, there is no 'pre-mining' step to find frequent (or maximal) subsequences. This allows us to apply different "projections" specializing the request of an expert and simplifying the computations. In addition, in our approach nearly all state-of-the-art FCA algorithms can be used in order to efficiently process a dataset.

There is a number of approaches that help to analyze medical treatment data. However, the direct comparison of them is hardly possible, because every approach is designed for its own problem. For example, (Tsumoto et al. 2014) analyze data of

one hospital and provide a different view on the processes within the hospital w.r.t. our approach. Finally and naturally, the most similar approach to our work can be found in (Egho et al. 2014a,b), as some authors of the present paper are involved in this alternative work. In (Egho et al. 2014a,b), authors mine frequent sequences of the dataset similar to the sequences studied here. However, they approach the complexity of the analysis of such data in a different way. They use a support threshold in order to specify the outcome of the algorithm and do not provide any order in which one can analyze the result. In our case we rely on projections that are usually simpler to incorporate expert knowledge than a support threshold and we give an order (w.r.t. stability of a concept) which can be used to simplify the analysis of the treatment data.

7. Conclusion

In this paper, we have presented a novel approach for analyzing sequential data within the framework of pattern structures, an extension of Formal Concept Analysis dealing with complex data. It is based on the formalism of sequential pattern structures and projections. Our work complements the general orientations towards *statistically significant patterns* by presenting strong formal results on the notion of interestingness from a concept lattice viewpoint. The framework of pattern structures is very flexible and shows some important properties, for example in allowing to reuse state-of-the-art and efficient FCA algorithms. Using pattern structures leads to the construction of a pattern concept lattice, which does not require the setting of a support threshold, as usually needed in classical sequential pattern mining. Moreover, the use of projections gives a lot of flexibility especially for mining and interpreting special kinds of patterns (patterns can be proposed at several levels of complexity w.r.t. extraction and interpretation).

Our framework was tested on a real-world dataset with patient hospitalization trajectories. Interesting patterns answering questions of an expert are extracted and interpreted, showing the feasibility and usefulness of the approach, and the importance of the stability as a pattern-selection procedure. In particular, projections play an important role here: mainly, they provide means to select patterns of a special interest and they help to save computational time (which could be otherwise very large).

For future work, we are planning to more deeply investigate projections, their potential w.r.t. the types of patterns. It can be interesting to introduce and evaluate the stability measure directly on sequences. Another research direction is mining of association rules or building a Horn approximation (Balcázar and Casas-Garriga 2005) from the stable part of the pattern lattice or stable sequences. Finally, as discussed above, a precise study combining frequent subsequence mining and FCA-based approaches should be carried out.

Acknowledgments

The fourth co-author was supported within the framework of the Basic Research Program at National Research University Higher School of Economics (Moscow).

Notes on contributors

Aleksey Buzmakov is a PhD student in Informatics at Université de Lorraine (Vandœvre de Nance, France). He holds master and bachelor degree in applied mathematics and physics from Moscow Institute of Physics and Technology. His research interest includes data mining and artificial intelligences. In particular he works with Formal Concept Analysis and Pattern Structure in order to mine complex data such as sequences or graphs.

Elias Egho is a Post Doctoral Researcher in Orange Labs (France Telecom Research and Development) with Profiling & Data Mining team. In 2014, he received a PhD degree in Computer Science from University of Lorraine, Nancy, France in LORIA-INRIA Nancy Grand Est laboratory. His main research interest is mining sequential patterns for detection and classification of sequential data.

Nicolas Jay is a professor of biostatistics and medical informatics at the Université de Lorraine. His research interests include medical knowledge representation and knowledge discovery in medical databases, with applications to patient trajectory analysis. He works as a public health physician at the University Hospital of Nancy.

Sergei O. Kuznetsov is a professor of the National Research University Higher School of Economics (HSE), Moscow, where he is the head of department of data analysis and artificial intelligence. He defended habilitation thesis ("Doctor of Science") at the Computer Center of the Russian Academy of Sciences (Moscow, Russia) in 2002. He holds the "Candidate of Science" degree (PhD equivalent) from VINITI (Moscow, Russia) since 1990. His research interests include mathematical models, algorithms and algorithmic problems of machine learning, formal concept analysis, data mining, and knowledge discovery.

Amedeo Napoli is a CNRS senior scientist (DR CNRS) and the scientific leader of the Orpailleur research team at LORIA/Inria Laboratory in Nancy. His scientific interests are knowledge discovery (pattern mining and Formal Concept Analysis) and knowledge representation (ontology engineering). He is involved in many national and international research projects with applications in agronomy, biology, chemistry, and medicine.

Chedy Raïsi received his PhD in Computer Science from the University of Montpellier and the Ecole des Mines d'Alès in July 2008. He is currently a research scientist ("Chargé de recherche 1") at the Institut "National de Recherche en Informatique et en Automatique" (INRIA) in France. His research interests includes pattern mining and privacy-preserving data analysis.

References

Adda, Mehdi, Petko Valtchev, Rokia Missaoui, and Chabane Djeraba. 2010. "A framework for mining meaningful usage patterns within a semantically enhanced web portal." In *Proceedings of the 3rd C* Conference on Computer Science and Software Engineering,* C3S2E '10. 138–147. New York, NY, USA: ACM.

Agrawal, Rakesh, and Ramakrishnan Srikant. 1995. "Mining Sequential Patterns." In *Proceedings of the Eleventh International Conference on Data Engineering,* ICDE '95. 3–14. Washington, DC, USA: IEEE Computer Society.

Ayres, Jay, Jason Flannick, Johannes Gehrke, and Tomi Yiu. 2002. "Sequential PAttern mining using a bitmap representation." In *KDD,* 429–435.

Balcázar, José L, and Gemma Casas-Garriga. 2005. "On Horn Axiomatizations for Sequential Data." In *ICDT,* 215–229.

Buzmakov, Aleksey, Elias Egho, Nicolas Jay, Sergei O. Kuznetsov, Amedeo Napoli, and Chedy Raïssi. 2013. "On Projections of Sequential Pattern Structures (with an application on care trajectories)." In *Proc. 10th International Conference on Concept Lattices and Their Applications,* 199–208.

Casas-Garriga, Gemma. 2005. "Summarizing Sequential Data with Closed Partial Orders.." In *Proc. of the 5th SIAM Int'l Conf. on Data Mining (SDM'05),* .

Chiu, Ding-Ying, Yi-Hung Wu, and Arbee L. P. Chen. 2004. "An Efficient Algorithm for

Mining Frequent Sequences by a New Strategy without Support Counting." In *ICDE,* 375–386.

Ding, Bolin, David Lo, Jiawei Han, and Siau-Cheng Khoo. 2009. "Efficient Mining of Closed Repetitive Gapped Subsequences from a Sequence Database." In *Proc. of IEEE 25th International Conference on Data Engineering,* 1024–1035. IEEE. Mar..

Egho, Elias, Nicolas Jay, Chedy Raïssi, Dino Ienco, Pascal Poncelet, Maguelonne Teisseire, and Amedeo Napoli. 2014a. "A contribution to the discovery of multidimensional patterns in healthcare trajectories." *Journal of Intelligent Information Systems* 42 (2): 283–305.

Egho, Elias, Chedy Raïssi, Nicolas Jay, and Amedeo Napoli. 2014b. "Mining Heterogeneous Multidimensional Sequential Patterns." In *ECAI 2014 - 21st European Conference on Artificial Intelligence,* 279–284.

Ferré, Sébastien. 2007. "The Efficient Computation of Complete and Concise Substring Scales with Suffix Trees." In *Formal Concept Analysis SE - 7,* Vol. 4390 of *Lecture Notes in Computer Science* edited by Sergei O. Kuznetsov and Stefan Schmidt. 98–113. Springer.

Fetter, R. B., Y. Shin, J. L. Freeman, R. F. Averill, and J. D. Thompson. 1980. "Case mix definition by diagnosis-related groups.." *Med Care* 18 (2): 1–53.

Ganter, Bernhard, and Sergei O. Kuznetsov. 2001. "Pattern Structures and Their Projections." In *Conceptual Structures: Broadening the Base,* Vol. 2120 of *Lecture Notes in Computer Science* edited by Harry S. Delugach and Gerd Stumme. 129–142. Springer Berlin Heidelberg.

Ganter, Bernhard, and Rudolf Wille. 1999. *Formal Concept Analysis: Mathematical Foundations*. 1st ed. Springer.

Han, Jiawei, Jian Pei, Behzad Mortazavi-Asl, Qiming Chen, Umeshwar Dayal, and Meichun Hsu. 2000. "FreeSpan: frequent pattern-projected sequential pattern mining." In *Proc. of the 6th ACM SIGKDD Int'l Conf. on Knowledge discovery and data mining,* 355–359.

Klimushkin, Mikhail, Sergei A. Obiedkov, and Camille Roth. 2010. "Approaches to the Selection of Relevant Concepts in the Case of Noisy Data." In *Proc. of the 8th International Conference on Formal Concept Analysis,* ICFCA'10. 255–266. Springer.

Kuznetsov, Sergei O. 1999. "Learning of Simple Conceptual Graphs from Positive and Negative Examples." In *Principles of Data Mining and Knowledge Discovery SE - 47,* Vol. 1704 of *Lecture Notes in Computer Science* edited by Jan M. Żytkow and Jan Rauch. 384–391. Springer Berlin Heidelberg.

Kuznetsov, Sergei O. 2007. "On stability of a formal concept." *Annals of Mathematics and Artificial Intelligence* 49 (1-4): 101–115.

Masseglia, Florent, Fabienne Cathala, and Pascal Poncelet. 1998. "The PSP Approach for Mining Sequential Patterns." In *PKDD,* 176–184.

Merwe, Dean Van Der, Sergei Obiedkov, and Derrick Kourie. 2004. "AddIntent: A new incremental algorithm for constructing concept lattices." In *Concept Lattices,* Vol. 2961 edited by Gerhard Goos, Juris Hartmanis, Jan Leeuwen, and Peter Eklund. 372–385. Springer.

Mooney, Carl H., and John F. Roddick. 2013. "Sequential pattern mining – approaches and algorithms." *ACM Computing Surveys* 45 (2): 1–39.

Pei, Jian, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu. 2001a. "PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth." In *ICDE,* 215–224.

Pei, Jian, Jiawei Han, B. Mortazavi-Asl, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu. 2001b. "PrefixSpan Mining Sequential Patterns Efficiently by Prefix Projected Pattern Growth." In *17th International Conference on Data Engineering,* 215–226.

Pinto, Helen, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, and Umeshwar Dayal. 2001. "Multi-Dimensional Sequential Pattern Mining." In *CIKM,* 81–88.

Plantevit, Marc, Anne Laurent, Dominique Laurent, Maguelonne Teisseire, and Yeow Wei Choong. 2010. "Mining multidimensional and multilevel sequential patterns." *ACM Transactions on Knowledge Discovery from Data* 4 (1): 1–37.

Raïssi, Chedy, Toon Calders, and Pascal Poncelet. 2008. "Mining conjunctive sequential patterns." *Data Min. Knowl. Discov.* 17 (1): 77–93.

Roth, Camille, Sergei Obiedkov, and Derrick G Kourie. 2008. "On succinct representation

of knowledge community taxonomies with formal concept analysis A Formal Concept Analysis Approach in Applied Epistemology." *International Journal of Foundations of Computer Science* 19 (02): 383–404.

Salvemini, Eliana, Fabio Fumarola, Donato Malerba, and Jiawei Han. 2011. "FAST sequence mining based on sparse id-lists." In *Proceedings of the 19th international conference on Foundations of intelligent systems,* ISMIS'11. 316–325. Berlin, Heidelberg: Springer-Verlag.

Srikant, Ramakrishnan, and Rakesh Agrawal. 1996. "Mining Sequential Patterns: Generalizations and Performance Improvements." In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology,* EDBT '96. 3–17. London, UK, UK: Springer-Verlag.

Tsumoto, Shusaku, Haruko Iwata, Shoji Hirano, and Yuko Tsumoto. 2014. "Similarity-based behavior and process mining of medical practices." *Future Generation Computer Systems* 33 (0): 21–31.

Yan, Xifeng, Jiawei Han, and Ramin Afshar. 2003. "CloSpan: Mining Closed Sequential Patterns in Large Databases." In *Proc. of SIAM Int'l Conf. Data Mining (SDM'03),* 166–177.

Yang, Zhenglu, Masaru Kitsuregawa, and Yitong Wang. 2006. "PAID: Mining Sequential Patterns by Passed Item Deduction in Large Databases." In *IDEAS,* 113–120.

Yu, Chung-Ching, and Yen-Liang Chen. 2005. "Mining Sequential Patterns from Multidimensional Sequence Data." *IEEE Trans. Knowl. Data Eng.* 17 (1): 136–140.

Zaki, Mohammed J. 2001. "SPADE: An Efficient Algorithm for Mining Frequent Sequences." *Mach. Learn.* 42 (1-2): 31–60.